VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

MATHS FOUNDATION for COMPUTER SCIENCE (055263)

Assignment

Community Structure Identification

(Version 0.2 for SEM222)

Instructors: Nguyen An Khuong, *CSE-HCMUT*
Nguyen Tien Thinh, *CSE-HCMUT*
Tran Tuan Anh, *CSE-HCMUT*

Teaching Assistants: Le Thanh Son, *Univ. of Limoges*
Tran Dinh Vinh Thuy, *Univ. of Limoges*

Ho Chi Minh, April 2023

Contents

1 Introduction

The origin of graph theory can be dated back to when Leonard Euler first gave his proof on the Königsberg seven bridges problem. By viewing the problem abstractly, using the notations of letters and lines, not only did Euler find the solution, he was able to generalize this problem. The idea of Euler on solving this problem put the stepping stones for the branch of graph theory. Since then, this branch has been an important field of mathematics, with much work made to study graphs and their mathematical properties. During the 20th and 21st century, there has been an explosion within the ways of using graphs to model real-life problems. Graphs are now becoming more and more useful to represent a wide variety of systems from different areas. For instance, a graph can be used to describe the relations of one person to others on a social network by treating the person as a vertex and each of their connections as an edge of a graph. Subsequently, a rising demand is to analyze the interactions among elements of a graph.

Besides using the normal representation of graphs, with edges and vertices, to deal with practical problems, we can take use of the description of graphs through matrices or tensors. Indeed, the duality between graph and matrix multiplication motivates the use of graph approaches in practice. Assuming we are performing a breadth-first search on a graph depicted in Figure 1. In the first step, we would like to find all the adjacent nodes of Alice, which are Bob and Carl on the graph. We recall that the relationship about adjacency between the nodes can be represented by an adjacency matrix $\mathbf{A} = (a_{ij})$ in which $a_{ij} = 1$ means that we have a connection from node i to node j and $a_{ij} = 0$ otherwise. The matrix $\mathbf{A}^\top$ (the transpose of $\mathbf{A}$) is shown in Figure 1 where the dots denote the values 1. Then, by using one-hot encoding vector $\mathbf{v}$ which is active at node Alice, we can observe that $\mathbf{A}^\top\mathbf{v}$ can be used to represent the set of adjacent nodes $\{\text{Bob}, \text{Carl}\}$.

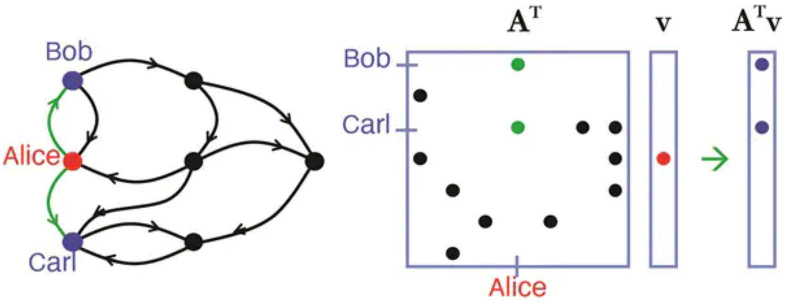

Figure 1: Duality between breadth-first search on a graph (left) and matrix multiplication (right) [1]

Graphical analysis has played a crucial part in understanding behaviors in various fields, such as biology, sociology, and computer science. Several problems are defined for each case of investigating such graph features.

2 Community structure identification problem

The one problem when using complex graphical networks to model real-life behaviors is that it tends to have a high level of order and organization within the elements of the graph. These graphs can display several clusters of density-connected vertices called communities. The vertices in one community probably share some common properties and/or behave similarly within the graph. One community is hence usually sparsely connected to other communities. Community structure identification in a graph aims to detect communities (see [2] for precise definition[1]), in other words, clusters, or possibly their hierarchy structure, given only the information of the graphical representation. It is one of the most important problems in graphical analysis (see [3] and in [4, Sections 10.2, 10.3, 10.5] for more detail), and it goes with many names in the literature: community detection, network clustering, graph partitioning, etc.

[1] See also at https://en.wikipedia.org/wiki/Community_structure and the references therein.

For example, in the social network scenario, consider a guy who works for an American start-up, goes to the tennis club every Friday and attends Chinese classes at a center. It is understandable to suppose that everyone in the start-up knows each other well and the same thing can be said for people from the tennis club and the Chinese classes. It is unlikely, however, to have almost every person in the start-up goes to the same center to learn some Chinese or goes to the same club to play tennis at the same time as the considering guy. This one particular instance illustrates that in real life, people tend to form community groups and behave accordingly in each group with little to no connection to other groups. Identifying such groups in the graphical representation enables the use of further applications.

3 Some use-cases

3.1 Social networks

When speaking of social networks, we think of Facebook, Instagram, or Twitter, to name a few. In these social networks, there are always relations between two entities residing in the network. On Facebook or Instagram, one account can follow or befriend another. Naturally, we want to represent the network of Facebook users as a graph, where each vertex represents one user. There will be an edge between two vertices if the two users represented by the vertices have a relation. This way of visualization can be enhanced if we want to represent more than just simple relationships by adding weights to the edges. For example, the “followed” edges will weigh 1 while 2 will be for the “befriended”. The communities in these networks will represent the groups that share things in common, such as being members of the same family or sharing a similar hobby. Identifying these communities is meaningful since we will have ideal targets to perform some specific operations. For example, we may have higher revenue by promoting hotel discounts to people from a traveling group than to those who are in a gaming group.

For further illustration, Figure 2 represents a simple example of a social network. There are three communities, each with a different number of members, represented by the large lines encircling the vertices. Also, we can observe that two of the communities are separated and two of them overlapped. In practice, we can have also a community that entirely belongs to another community.

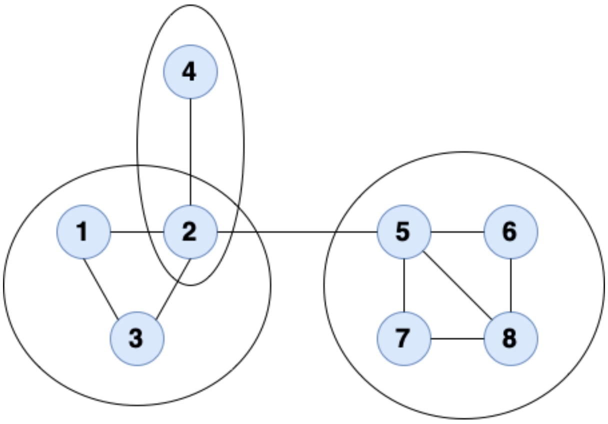

Figure 2: Social network using graphical representation

Besides social networks as discussed, there are numbers of other use cases that can be represented in the same way such as the followings.

3.2 Customer networks

Suppose that we are running an online bookstore. Customers may be interested in more than one type of book. They can find their interests in the latest fantasy thriller by Stephen King while sharing some of their pleasures for mathematical textbooks. Finding which types of books each customer invests in can help us improve our service through investigations in the detected communities. In this case, each book is represented by a vertex in a graph and an edge between two books is created if these two are bought by the same customer.

3.3 Collaboration networks

Consider an interdisciplinary research center located in Ho chi Minh City. We can construct a graph whose vertices represent scientists in residence during any time window, say from 2013 to 2023. An edge will be created between two vertices if the corresponding researchers appear in at least one joint work during the given time. The communities will represent the people working on the same particular topic or with similar methodologies. In this use-case, we can see that there can be communities within a community and overlap communities. For instance, a research article proposing a new algorithm can be coauthored by mathematicians for the theoretical parts and computer scientists for the experimental sections. Besides, we can have various authors working in the same field, say computer vision, but they can form into more specific tasks such as image denoising, object detecting, etc. Finding such communities within the constructed graph can help tracing related works less burden, which is sufficient in the first step of research.

3.4 Transportation networks: Multi-link graphical representation

In cities around the world, there are normally means of transportation to go from one city to another, such as car, plane, or ship. These are considered to be relationships in a graph where each city is a node. Hence, we place one edge for each line between the two cities. In this case, the graph is more challenging to be analyzed compared to the one introduced in Subsection 3.1, which is called *multi-link graph* [5, Section 7.1]. We are interested in finding the communities in this graph so we can determine the similarities between cities, which help us in traveling or shipment services. Figure 3 illustrates a multi-link transportation graph with two detected communities. There is more than one link between City 1 and City 2, hence, our graph is a multi-link graph. To represent this graph, we can use the adjacent tensor.

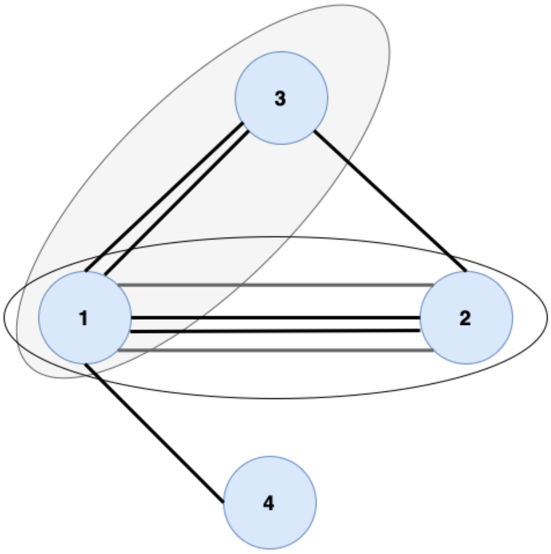

Figure 3: Multi-link graph representation

As in Section 1, we also have the duality between the multi-link graph and the tensor. Thus, besides some classical methods in [6, 7, 8], we can perform *tensor decomposition* described in [5, Chapter 7].

Tensor decomposition methods such as *principal component analysis*[2] or *low-rank approximation*[3] could also give insights about the latent structure.

4 Traditional approaches to the community structure identification problem

We introduce some approaches that can be applied in the community structure identification problem. These are to get you the ideas of the approaches. You are encouraged to choose your preference method.

4.1 Hierarchical clustering

Since community structure identification sometimes goes with the name of graph clustering, we can apply some traditional graph clustering approaches such as hierarchical clustering (see Chapter 7 in [4]). In this approach, the communities are created by the "closeness" between the vertices in the graph. The definition of "closeness" may vary, depending on the particular algorithm.

4.2 Girvan-Newman algorithm

The Girvan-Newman algorithm [9, 10] detects communities by removing the edges iteratively according to their highest "betweenness" score. This algorithm is based on the intuition we discussed in Section 2 that the communities are sparsely connected by a few edges located in the shortest path between vertices. Thus, removing these edges will reveal the communities inside the graph. Some other enhancements of this algorithm can be founded in [11] and [12].

4.3 Louvain algorithm

The idea of the Louvain algorithm [13] is based on the maximization of "modularity", which is a value measuring the density of edges inside a community. At each iteration, the algorithm places the vertices to other communities until to attain maximum modularity and it stops when there is no further improvement in modularity. There are also many improvements to this algorithm that can be found on the Internet.

4.4 Matrix factorization-based methods

If we used adjacent matrix to represent the graph, we can apply the non-negative matrix factorization methods (MNF) on the adjacent matrix to find the communities within the graph. The general idea of this method is to factorize the adjacent matrix $\mathbf{A}$ by two smaller matrices $\mathbf{W}$ and $\mathbf{H}$ such that $\mathbf{A} \approx \mathbf{WH}$ and $\mathbf{W}$ and $\mathbf{H}$ both have no negative elements. By examining the structures of $\mathbf{W}$ and $\mathbf{H}$, we can find identify the communities within the graph. Some particular research works on this approach can be found in [14], [15], or [16].

4.5 Tensor factorization-based methods

When working with a multi-link graph, we can not use an adjacent matrix. Instead, we must use a tensor to represent our graph. The traditional matrix factorization-based approaches obviously will not work in this case. Hence, we will consider ways to factorize the tensor representing the graph. The CP decomposition [17, Section 3] can be considered as a generalization of the SVD for tensor. Indeed, with the given rank R, we can decompose a tensor $\boldsymbol{\mathcal{X}}$ as the sum of R rank-one tensors. Precisely, each factor represents a "community" within the data and the number of factors R in the approximation should loosely reflect the number of communities in the data [5, Subsection 7.2.4]. There also exists other means of tensor decomposition, such as Tucker decomposition [17, Section 4]. Examinating the structures generated by these decompositions can help finding and understanding the underlying communities of a graph.

[2] https://en.wikipedia.org/wiki/Principal_component_analysis

[3] https://en.wikipedia.org/wiki/Low-rank_approximation

5 Suggested datasets

We now give some datasets that can be used for the experimental parts of this assignment, note that we do not state the explicit use case for these datasets and some of these datasets require additional preprocessing before being able to put into your version of implementations.

- Dolphins online social network: A social network of bottlenose dolphins. The dataset contains a list of all of links, where a link represents frequent associations between dolphins.
- Political books: A network of books about US politics published around the time of the 2004 presidential election and sold by the online bookseller amazon.com. Edges between books represent frequent co-purchasing of books by the same buyers.
- Jazz musicians: A network between Jazz musicians. Each node is a Jazz musician and an edge denotes that two musicians have played together in a band.
- NIPS Conference Papers Vols 0-12: A dataset contain full information of papers published from volume 0 to 12 of conference on neural information processing systems (NIPS). This dataset requires preprocessing to create a meaningful graphical representation, for instance, the collaboration networks as in Section 3.

Several other datasets can be found here and here, each with corresponding description and download link. Other datasets can also be found in the content of the papers in the References.

If you find the given suggestions hard to interpret or unsatisfied, you can find from other sources that are not listed here or created your synthesis data. However, if you opt for this direction, please indicate specifically where you found the dataset on the internet or how your dataset created , its description (number of vertices and edges, some other prevelence statistics,...).

6 Guidelines

6.1 Objectives (Questions)

1. Choose a use-case that requires or can apply the idea of community structure identification to solve (be careful in setting the problem and limiting the scope so that your team can manage to solve it within 4 weeks). You can refer to some ideas in Section 3, and in fact, you can find dozens of use cases related to the community structure identification problem through the references in this assignment. You are encouraged to find or construct one use case of your own if you find the given suggestions unsatisfied.
2. With the chosen use case, find or create a suitable dataset that best represents your problem, see Section 5 for more details. There will be a penalty if your choice dataset is not specified.
3. Study and implement your chosen approach for this problem and experiment with your dataset (see [18] for some of the most popular approaches, and see [19, Section 7.2] for sample codes in R or Python).
4. Explain in detail the behaviors and the results of your approach, especially the motivation, the approaches, the metrics, and the dataset you use.

6.2 Requirements and Instructions

1. Form your group of up to 5 people and at least four at `https://e-learning.hcmut.edu.vn/mod/groupselect/view.php?id=135893`. The group list must be updated before Aprile 20, 2023.
2. Write a technical report (advisedly in English if possible) by LaTeX[4] justifying the group's use-case by modeling the problem using any graphical representation and specifying what communities mean in your case. This report should also explain your data choices and approach

[4] See templates and sample available at `https://www.overleaf.com/gallery/tagged/report`

in-depth and give meaningful observations and interpretations of your results. The report should also cite all of your references, whether it is an article, a research paper, a textbook, etc. The structure of the report can be the following

(a) **Chapter I. Introduction**: Explain your motivation, and introduce your use-case with a problem statement (what you are trying to do with this use-case) on a specific dataset.

(b) **Chapter II. Preliminaries**: Recall or present all the definitions and properties with concrete examples of all foundations for later uses. [*Doing this part carefully will be helpful with your final examination.*]

(c) **Chapter III. Approaches**: Present your choices of algorithm or method that can be used to solve the problem with a detailed explanation.

(d) **Chapter IV. Experiments**: Perform numerical experiments using the implemented approach or approaches on your dataset, and give your in-depth analysis.

(e) **Chapter V. Conclusion**: Summarize what you have done, your limitations, and the directions for future works.

(f) **References**.

3. Write the codes (required) and prepare demonstrations (if any) your group made during the time doing this assignment.

4. Prepare slides by Beamer[5] explaining your group work thoroughly. The number of slides should not be lengthy (at most 20 slides, preferably 15 slides or less) with adequate contents of your work. All groups must give presentations with a reporting time of no more than 15 minutes for each group during the two last lectures (scheduled as make-up lectures and will take place on May 15-21, 2023). You will be provided with the entire content of all other groups before the presentation day. Based on that, each group must prepare at least two questions and two comments/suggestions for all other groups to give the presentation on another day.

5. All of the meeting minutes of your group must be combined into one .txt file and consist of the % of each member's effort on the whole (total effort is 100%) on the first line.

6. Please compress all materials relating to your work as mentioned above in **one .zip file** and **only the team leader submits** it to the e-learning site of this course.[6] You will have **4 weeks to do this project, starting from April 17, 2023** (hard deadline, there will be no extension.) Note that plagiarism is strictly prohibited and will be handled accordingly.

7. Some basic technical backgrounds related to this assignment will be asked in the final exam. Therefore, team members must work together so that all of you understand all aspects of the project. The team leader should organize the team to meet this requirement.

This assignment is two-fold:

1. You are expected to find at least an applicable use-case that can be modeled by a single graph for the community structure identification problem, such as the ones introduced in Section 3.1, and implement at least one algorithm from the suggestions in Section 4 to solve this problem with your use case.

2. At a more advanced level, you can work a use-case where a multi-graph must be used to model and implement some appropriate algorithms to find the answers to your problem. You can also tackle a case where the graph's adjacent matrix (or tensor) representation is sparse. If you opt for this direction, you can work with a group of less than 4 (strong) members.

If you are not used to working with graphs or/and having trouble comprehending concepts in English. In that case, a suggestion is to support with [3], rather well-written in Vietnamese and provided with several explicit computational examples, to grasp the main idea of this assignment before continuing working.

[5] See templates and samples available at https://www.overleaf.com/gallery/tagged/presentation

[6] https://e-learning.hcmut.edu.vn/mod/assign/view.php?id=57018

References

[1] J. Kepner, D. A. Bader, T. Davis, R. Pearce, and M. M. Wolf, “Graphblas and graphchallenge advance network frontiers,” *SIAM News*, vol. 55, no. 08, October 2022, URL: https://sinews.siam.org/Details-Page/graphblas-and-graphchallenge-advance-network-frontiers.

[2] S. Fortunato and C. Castellano, *Community Structure in Graphs*, pp. 490–512. New York, NY: Springer New York, 2012, URL: https://doi.org/10.1007/978-1-4614-1800-9_33.

[3] N. T. Duc, “Community detection on social graphs, *B. Eng Thesis, HCMUT, VNU-HCM* (in vietnamese),” 2018, URL: https://link.gdsc.app/b8W2A87.

[4] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive data sets*. Cambridge University Press, 2020, URL: https://www.mmds.org/.

[5] J. Kepner and J. Gilbert, *Graph algorithms in the language of linear algebra*. SIAM, 2011, URL: https://doi.org/10.1137/1.9780898719918.

[6] J. Kim and J.-G. Lee, “Community detection in multi-layer graphs: A survey,” *ACM SIGMOD Record*, vol. 44, no. 3, pp. 37–48, 2015.

[7] L. Getoor and C. P. Diehl, “Link mining: a survey,” *Acm Sigkdd Explorations Newsletter*, vol. 7, no. 2, pp. 3–12, 2005.

[8] L. Getoor, N. Friedman, D. Koller, and B. Taskar, “Learning probabilistic models of link structure,” *Journal of Machine Learning Research*, vol. 3, no. Dec, pp. 679–707, 2002.

[9] M. Girvan and M. E. Newman, “Community structure in social and biological networks,” *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002, URL: https://www.pnas.org/doi/10.1073/pnas.122653799.

[10] M. E. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical review E*, vol. 69, no. 2, p. 026113, 2004, URL: https://arxiv.org/pdf/cond-mat/0308217.pdf.

[11] M. E. Newman, “Fast algorithm for detecting community structure in networks,” *Physical review E*, vol. 69, no. 6, p. 066133, 2004, URL: https://arxiv.org/pdf/cond-mat/0309508.pdf.

[12] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, “Defining and identifying communities in networks,” *Proceedings of the national academy of sciences*, vol. 101, no. 9, pp. 2658–2663, 2004, URL: https://www.pnas.org/doi/10.1073/pnas.0400054101.

[13] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008, URL: https://arxiv.org/pdf/0803.0476.pdf.

[14] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, “Community discovery using nonnegative matrix factorization,” *Data Mining and Knowledge Discovery*, vol. 22, no. 3, pp. 493–521, 2011, URL: https://www.researchgate.net/publication/220451825_Community_discovery_using_nonnegative_matrix_factorization.

[15] Z.-Y. Zhang, Y. Wang, and Y.-Y. Ahn, “Overlapping community detection in complex networks using symmetric binary matrix factorization,” *Physical Review E*, vol. 87, no. 6, p. 062803, 2013, URL: https://arxiv.org/pdf/1303.5855.pdf.

[16] N. P. Nguyen and M. T. Thai, “Finding overlapped communities in online social networks with nonnegative matrix factorization,” in *MILCOM 2012-2012 IEEE Military Communications Conference*, pp. 1–6, IEEE, 2012, URL: https://ieeexplore.ieee.org/abstract/document/6415744.

[17] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[18] V. L. Dao, C. Bothorel, and P. Lenca, "Community structure: A comparative evaluation of community detection methods," *Network Science*, vol. 8, no. 1, pp. 1–41, 2020, URL: https://arxiv.org/pdf/1812.06598.

[19] K. McNulty, *Handbook of Graphs and Networks in People Analytics: With Examples in R and Python*. CRC Press, 2022, URL: https://ona-book.org/.